IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 August 2022, accepted 24 September 2022, date of publication 30 September 2022,
date of current version 10 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3210974

RESEARCH ARTICLE

Time Series Forecasting Based on Empirical Mode Decomposition and the Varying-Coefficient DBN-AR Model

HUI HU[1] AND WENQUAN XU[2]
[1]School of Teacher Education, Anqing Normal University, Anqing, Anhui 246001, China
[2]School of Electronic Engineering and Intelligent Manufacturing, Anqing Normal University, Anqing, Anhui 246001, China

Corresponding author: Wenquan Xu (zyheijunma@163.com)

This work was supported in part by the Key Projects of Natural Science Research in Colleges and Universities of Anhui Province under Grant KJ2020A0508, in part by the Humanities and Social Sciences Research Project of Anhui Province under Grant SK2021A0374, and in part by the Key Research and Development Plan of Anhui Province under Grant 202104a05020050.

ABSTRACT In an attempt to improve the prediction accuracy for a time series, a new one-step-ahead hybrid model is proposed in this paper that combines empirical mode decomposition (EMD) with the DBN-AR model and back propagation (BP). The proposed approach first uses EMD, which can be used to decompose the complicated original time series data into several intrinsic mode functions (IMFs) and a residue. The IMF components and residue are then modelled and forecasted using the DBN-AR model. Finally, the predicted results for all IMFs and a residue are combined by a BP neural network to obtain an aggregated output for the time series data. To evaluate the performance of the proposed hybrid model, Beijing PM2.5 level time series data and the weekly rates of British Pound/US dollar (GBP/USD) exchange rate data are used as an illustrative example. Experimental results demonstrate the attractiveness of the proposed hybrid model based on both the prediction accuracy and efficiency compared with other methods.

INDEX TERMS Time series forecasting, EMD, deep belief network, hybrid method, decomposition and ensemble framework.

I. INTRODUCTION

In recent years, time series prediction has become a very hot research topic. The analysis of a time series is meaningful for research. The prediction of traffic conditions can help one to arrange travel. Mankind can avoid disasters by predicting disasters in advance. However, different time series data contain different characteristics [1]. Some data are highly volatile, for example, wind speed data series, and some data are less volatile, such as annual rainfall [1]. Some data are linear in nature, such as human beings, but most data series are nonlinear in nature [1]. For research into time series, several challenges still need to be addressed, including prediction accuracy for time series. Historical observations of the same variable are analysed to establish a model to describe a potential relationship by the time series prediction method. However, it is difficult to use the same model to predict different time series [2]. Many prediction models or methods have been proposed in the literature for time series forecasting, which can be classified into three categories: statistical models, artificial intelligence (AI) models and hybrid modes [3]. In the first category, the widely used models mainly include autoregressive (AR), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA) models. In [4], an AR model is used to predict the daily 10.7 cm solar radio flux, and the experimental results show that the AR model performs well. Zhang *et al.* [5] predicted particulate matter time series in Taiyuan, China by using ARIMA and ARMA models. The results show that the traditional ARMA/ARIMA can reduce the forecasting error. In fact, real-world time series data are often nonlinear, therefore, it is inappropriate to use such linear prediction models when the original time series data is nonlinear.

The associate editor coordinating the review of this manuscript and approving it for publication was Sajid Ali.

In the second category, many AI models have been proposed to predict time series, such as artificial neural networks (ANNs), support vector machines (SVMs) and deep belief networks (DBNs). For example. Khaniani *et al.* [6] used a neural network model to predict rainfall, and the results showed that the ANN is highly efficient. In [7], a SVM is used to predict financial series, and the experiments show that the SVM is a useful prediction model. Hu *et al.* [8] predicted the remaining useful life based on a DBN, and the experiment showed that the DBN model is effective and superior. Shan *et al.* [9] proposed a new neural network model for mathematical expression recognition. Xun *et al.* [10] proposed a dynamic ANN model called the meta-ANN for forecasting short-term grid loads, and a numerical study showed that the proposed model can be used to obtain more accurate and robust predictions for grid loads.

However, the original data series predicted by traditional single prediction models cannot be used to describe the complicated relations existing in the data series. Therefore, many hybrid models have been proposed by researchers. For example, in [11], a hybrid local linear neuro-fuzzy model is proposed, and used for nonlinear time series forecasting. A comparison of the results shows the superiority and promising performance of the proposed hybrid approach. Diao *et al.* [12] proposed a hybrid model for short-term traffic volume prediction, and the simulation results demonstrate that their hybrid model can be used to obtain better prediction accuracy. In [13], neural network and multiobjective optimization combined with a novel hybrid model was used to predict effective load data series, and the experimental results showed that the hybrid model can outperform baseline models.

There is also a special hybrid model, and different data decomposition techniques can be used to predict time series data before forecasting [14], [15]. For example, Wang *et al.* [16] proposed a hybrid model based on a two-layer decomposition technique and a BP neural network for multistep ahead electricity price forecasting, and the experiment demonstrated that the proposed model has superior electricity price performance. In [17], Qiu *et al.* proposed a hybrid approach composed of a discrete wavelet transform (DWT), empirical mode decomposition (EMD) and random vector functional link network (RVFL) for short-term electric load forecasting, and the experiment verified the effectiveness of the proposed method.

The aim of this paper is to develop an EMD-based DBN-AR model and BP neural network to predict time series with high accuracy. EMD can be used to analyze the nonlinear and non-stationary properties of time series. EMD is used to decompose the original time series into several modes [18]. The obtained time series for each decomposed mode is predicted by the DBN-AR model, and then the prediction results for all intrinsic mode functions (IMFs) and a residue are combined by a BP neural network to obtain an aggregated output for the time series data. The prediction results of this paper have demonstrated that the proposed hybrid model outperforms the singel model for the PM2.5 time series and the weekly rates of British Pound/US dollar (GBP/USD) exchange rate data.

The rest of this paper is organized as follows: Section 2 briefly introduces the EMD. The nonlinear DBN-AR model is listed in Section 3. The basic framework of the proposed hybrid model is given in Section 4. Section 5 gives evaluation indicators. The experimental prediction results for the PM2.5 time series dataset and the weekly rates of British Pound/US dollar (GBP/USD) exchange rate data are demonstrated in Section 6. Finally, conclusions are stated in Section 7.

II. EMPIRICAL MODE DECOMPOSITION

In 1998, Huang *et al.* proposed the empirical mode decomposition (EMD) method, which decomposes the original data series into several intrinsic mode functions (IMFs) and a residue [18]. Instantaneous frequency data from nonstationary and nonlinear data series can be obtained by the EMD method. Two criteria must be satisfied by each IMF component, one is that the difference between the number of crossings times and the number of local extreme values is at most one; the other is that the mean value of the upper and lower envelopes of an IMF is zero at any time, which is identified by local maxima and minima. For a given original data series $\upsilon(t)$, the decomposition steps for EMD are given as follows:

(1) For a given original data series $\upsilon(t)$, a cubic-spline interpolation of the local minima and maxima is used to create the lower and upper envelopes for the data series.

(2) The value $m_1(t)$ is the mean of the upper and lower envelopes.

(3) The value of mean $m_1(t)$ is subtracted from the original data series $\upsilon(t)$, to obtain a removed low frequency data series. After subtracting the value of the mean from the value of the original data, a removed low-frequency time series is obtained $h_1(t) = \upsilon(t) - m_1(t)$.

(4) Suppose $h_1(t)$ is an IMF; then, $h_1(t)$ is the first component of the original data series $\upsilon(t)$, and steps (1) to (3) are repeated until the stopping criteria are satisfied. Therefore, the first IMF component $c_1(t)$ is obtained $c_1(t) = h_1(t)$.

(5) The original data series $\upsilon(t)$ is subtracted from the first component $c_1(t)$, and the residue signal is computed by $r_1(t) = \upsilon(t) - c_1(t)$.

(6) The obtained residual signal $r_1(t)$ is used as a new original data series $\upsilon(t)$ to find the next IMF. Steps (1) to (5) are repeated until the last residual data series becomes a monotonic function.

Finally, the original data series signal $\upsilon(t)$ is decomposed as follows

$$\upsilon(t) = \sum_{i=1}^{n} c_i(t) + \varpi_n(t) \tag{1}$$

where n is the number of IMFs, $c_i(t)$ is the IMFs and $\varpi_n(t)$ is the symbolized residue.

III. DEEP BELIEF NETWORK-BASED VARYING-COEFFICIENT AUTOREGRESSIVE MODEL

A. STRUCTURE OF THE DBN-AR MODEL

A DBN is stacked by several restricted Boltzmann machines (RBMs) as proposed by Hinton [19]. For each RBM, each subnetwork's hidden layer is the visible layer for the next RBM. Finally, a logistic regression is added to the top of the stack [20]. A schematic of a DBN is given in Fig. 1. The input-output relationship for the DBN model is shown in Eq. 2.

$$
\begin{cases}
\vartheta\left(\boldsymbol{\Theta}(t-1)\right)=\varphi\left(\mathbf{w}_1^{(N_r)}\mathbf{h}^{(N_r-1)}(t)+b_1^{(N_r)}\right)\\
\mathbf{h}^{(\ell)}(t)=\left(h_1^{(\ell)}(t),h_2^{(\ell)}(t),\cdots,h_{Q_\ell}^{(\ell)}(t)\right)^{\mathrm{T}},\\
\qquad \ell\in\{1,2,\cdots,N_r-1\}\\
h_{n_\ell}^{(\ell)}(t)=\varphi\left(\mathbf{w}_{n_\ell}^{(\ell)}\mathbf{h}^{(\ell-1)}(t)+b_{n_\ell}^{(\ell)}\right),\\
\qquad n_\ell\in\{1,2,\cdots,Q_\ell\}\\
\mathbf{w}_{n_\ell}^{(\ell)}=\left(w_{n_\ell,1}^{(\ell)},w_{n_\ell,2}^{(\ell)},\cdots,w_{n_\ell,Q_{\ell-1}}^{(\ell)}\right),\\
\qquad Q_0=n_y\\
\mathbf{h}^{(0)}(t)=\boldsymbol{\Theta}(t-1)
\end{cases}
\tag{2}
$$

where $\mathbf{w}_{n_\ell}^{(\ell)}$ is the parameter weight between the ℓ-th layer and the $(\ell-1)$-th layer in the single DBN model, $\left(b_1^{(\ell)},b_2^{(\ell)},\cdots,b_{Q_\ell}^{(\ell)}\right)$ is the bias in layer ℓ, n_y is the output lag, and Q_ℓ represents the number of nodes on layer $\ell\cdot N_r$ represents the total number of layers, $\mathbf{h}^{(\ell)}(t)$ represents the output of the ℓ-th layer, $\vartheta(\boldsymbol{\Theta}(t-1))$ represents the output of the single DBN model, $\varphi(x)=1/\left(1+e^{-x}\right)$ is a sigmoid function, and $\boldsymbol{\Theta}(t-1)$ is the input state vector of the DBN model.

In general, when a nonlinear system is considered, Eq. 3 is used to describe the nonlinear system:

$$
y(t)=f\left(y(t-1),y(t-2),\ldots,y\left(t-n_y\right)\right)+\varepsilon(t)
\tag{3}
$$

where $f(\bullet)$ represents a nonlinear map, $y(t)\in\Re$ is the output, n_y is the output lag, and $\varepsilon(t)\in\Re$ denotes the noise.

In this paper, a state-dependent autoregressive (SD-AR) model is used to approximate the nonlinear mapping $f(\bullet)$ in Eq. (3), which gives the following expression:

$$
y(t)=\gamma_0\left(\boldsymbol{\Theta}(t-1)\right)+\sum_{m=1}^{n_y}\gamma_{y,m}\left(\boldsymbol{\Theta}(t-1)\right)y(t-m)+\xi(t)
\tag{4}
$$

where $\gamma_0(\boldsymbol{\Theta}(t-1))$ and $\gamma_{y,m}(\boldsymbol{\Theta}(t-1))(m=1,\ldots,n_{y-1},n_y)$ are the state-dependent function type coefficients of the SD-AR model (4). $\boldsymbol{\Theta}(t-1)=y(t-1),y(t-2),\ldots,y\left(t-n_y\right)$ represents the state vector at time t, which is the variable used to change the system's working point with time; in some cases, this can be the output data series and/or the input data series.

A set of DBNs are used to approximate the function-type coefficients of the SD-AR model (4), and then the DBN-AR model is obtained [2]. The input-output relationship for the DBN-AR model can be described as in Eq. (5), shown at the bottom of the page. The DBN-AR model (5) is used to predict each IMF components and residue in this paper. It also can be seen from Eq. (5) that the DBN-AR model (5) is built as a global model, and it can be chosen as a locally linear AR model when $\boldsymbol{\Theta}(t-1)$ in Eq. (5) is fixed at time t. Furthermore, the complexity of the model (5) can be partially dispersed into the AR part of the model, where $\mathbf{w}_{n_{\ell_j}^{(j)},j}^{(\ell_j)}$ represents a weight matrix connecting layer ℓ_j and layer ℓ_j-1 in the j-th DBN module, $Q_{\ell_j}^{(j)}$ denotes the number of nodes on layer ℓ_j in the j-th DBN module, $\mathbf{h}_j^{(\ell_j)}(t)$ denotes the output of the ℓ_j-th hidden layer in the j-th DBN module, $\left(b_{1,j}^{(\ell_j)},b_{2,j}^{(\ell_j)},\cdots,b_{Q_{\ell_j}^{(j)},j}^{(\ell_j)}\right)$ are

$$
\begin{cases}
y(t)=\gamma_0\left(\boldsymbol{\Theta}(t-1)\right)+\sum_{m=1}^{n_y}\gamma_{y,m}\left(\boldsymbol{\Theta}(t-1)\right)y(t-m)+\xi(t)\\
\gamma_0\left(\boldsymbol{\Theta}(t-1)\right)=\varphi\left(u_{1,0}^{\left(N_r^{(0)}\right)}(t)\right)=\varphi\left(\mathbf{w}_{1,0}^{\left(N_r^{(0)}\right)}\mathbf{h}_0^{\left(N_r^{(0)}-1\right)}(t)+b_{1,0}^{\left(N_r^{(0)}\right)}\right)\\
\gamma_{y,m}\left(\boldsymbol{\Theta}(t-1)\right)=\varphi\left(u_{1,m}^{\left(N_r^{(m)}\right)}(t)\right)=\varphi\left(\mathbf{w}_{1,m}^{\left(N_r^{(m)}\right)}\mathbf{h}_m^{\left(N_r^{(m)}-1\right)}(t)+b_{1,m}^{\left(N_r^{(m)}\right)}\right),\quad m\in\left\{1,2,\cdots,n_y\right\}\\
\mathbf{h}_j^{(\ell_j)}(t)=\left(h_{1,j}^{(\ell_j)}(t),h_{2,j}^{(\ell_j)}(t),\cdots,h_{Q_{\ell_j}^{(j)},j}^{(\ell_j)}(t)\right)^{\mathrm{T}},\qquad \ell_j\in\left\{1,2,\cdots,N_r^{(j)}-1\right\},\ j\in\left\{0,1,2,\cdots,n_y\right\}\\
h_{n_{\ell_j}^{(j)},j}^{(\ell_j)}(t)=\varphi\left(u_{n_{\ell_j}^{(j)},j}^{(\ell_j)}(t)\right)=\varphi\left(\mathbf{w}_{n_{\ell_j}^{(j)},j}^{(\ell_j)}\mathbf{h}_j^{(\ell_j-1)}(t)+b_{n_{\ell_j}^{(j)},j}^{(\ell_j)}\right),\qquad n_{\ell_j}^{(j)}\in\left\{1,2,\cdots,Q_{\ell_j}^{(j)}\right\}\\
\mathbf{w}_{n_{\ell_j}^{(j)},j}^{(\ell_j)}=\left(w_{n_{\ell_j}^{(j)},1,j}^{(\ell_j)},w_{n_{\ell_j}^{(j)},2,j}^{(\ell_j)},\cdots,w_{n_{\ell_j}^{(j)},Q_{\ell_j-1}^{(j)},j}^{(\ell_j)}\right),\ Q_0^{(j)}=n_y\\
\mathbf{h}_j^{(0)}(t)=\boldsymbol{\Theta}(t-1)=\left(y(t-1),y(t-2),\ldots,y\left(t-n_y\right)\right)^{\mathrm{T}}\\
\qquad=\left(\kappa_1,\kappa_2,\cdots,\kappa_{n_w}\right)^{\mathrm{T}}
\end{cases}
\tag{5}
$$

the biases of the ℓ_j-th hidden layer in the j-th DBN module, and $\gamma_0\left(\mathbf{\Theta}(t-1)\right), \gamma_{y,m}\left(\mathbf{\Theta}(t-1)\right), \left(m=1,2,\cdots,n_y\right)$ is the output of each DBN module in the DBN-AR model (5), which corresponds to the state-dependent function-type coefficients. $\left(\kappa_1,\kappa_2,\cdots,\kappa_{n_w}\right)^{\mathrm{T}}$ denotes the input vector, and n_w is the dimension of $\mathbf{\Theta}\left(t-1\right)$.

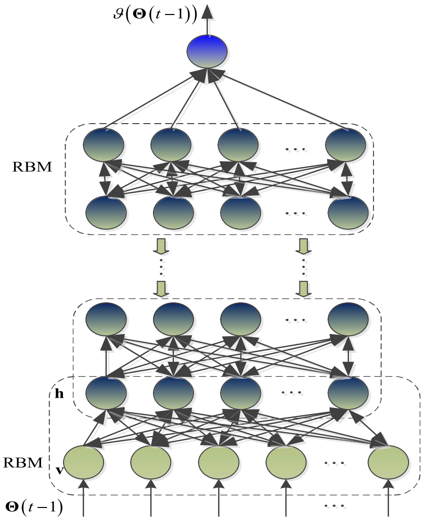

FIGURE 1. Structure of a single DBN model.

B. ESTIMATION OF THE DBN-AR MODEL

First, the original data series $y\left(t\right)$ is normalized to $y^{\circ}\left(t\right) = \frac{y(t)-\min(y(t))}{\max(y(t))-\min(y(t))}$, and the value of $y^{\circ}\left(t\right)$ is scaled between 0 and 1. Then the reference output data series for $\gamma_0\left(\mathbf{\Theta}(t-1)\right)$ and $\gamma_{y,m}\left(\mathbf{\Theta}(t-1)\right)\left(m=1,2,\cdots,n_y\right)$ are assigned for each DBN module in the DBN-AR model (5) based on the pseudo inverse matrix and the least squares solution. For the normalized data series $\{y^{\circ}\left(i\right), y^{\circ}\left(i+1\right), \cdots, y^{\circ}\left(n_y+i\right)\}$, $i\in\left\{1,2,\cdots,N-n_y\right\}$, according to the DBN-AR model (5) based on using the least squares solution the reference outputs of $\left\{\gamma_0(\mathbf{\Theta}(t-1)), \gamma_{y,1}(\mathbf{\Theta}(t-1)), \gamma_{y,2}(\mathbf{\Theta}(t-1)), \cdots, \gamma_{y,n_y}(\mathbf{\Theta}(t-1))\right\}$ in the DBN modules at sample instant n_y+i-1 are calculated. Eq. (6) is used to calculate the output of the DBN-AR model (5):

$$y^{\circ}(n_y+i) = \mathbf{M}_{n_y+i-1}\Phi_{n_y+i-1}, i\in\left\{1,2,\cdots,N-n_y\right\} \quad (6)$$

where, as shown in the equation at the bottom of the page, Φ_{n_y+i-1} represents the reference output of the DBN modules, and $\mathbf{M}_{n_y+i-1}$ is the coefficient in model (6) at time n_y+i-1. The target value of the DBN modules in the DBN-AR model (6) can be calculated from

$$\Phi_{n_y+i-1} = \mathbf{M}^{+}_{n_y+i-1}y^{\circ}(n_y+i) \quad (7)$$

where $\mathbf{M}^{+}_{n_y+i-1}$ denotes the pseudo inverse matrix of $\mathbf{M}_{n_y+i-1}$. Next, the parameters for each DBN module are trained by the deep learning algorithm using the reference output from the DBN modules computed by (7) in this pretraining stage. After this pretraining, the prediction output of the DBN-AR model (5) is calculated using Eq. (8) in this pretraining stage.

$$\begin{aligned}\widehat{y}(t) &= \hat{\gamma}_0\left(\mathbf{\Theta}(t-1)\right) + \sum_{m=1}^{n_y}\hat{\gamma}_{y,m}\left(\mathbf{\Theta}(t-1)\right)y^{\circ}(t-m),\\ t &= n_y+1, n_y+2, \cdots, N \end{aligned} \quad (8)$$

where $\widehat{y}(t)$ is the predicted output value of the DBN-AR model (5) in the pretraining stage, and the output values, $\left(\hat{\gamma}_0\left(\mathbf{\Theta}(t-1)\right), \hat{\gamma}_{y,m}\left(\mathbf{\Theta}(t-1)\right)(m=1,2,\cdots,n_y)\right)$, of each DBN module are also obtained from model (6) after pretraining.

Finally, an especially designed back propagation (BP) algorithm (see Appendix) is used to fine tune the parameters of the DBN-AR model (6). After pretraining, the modelling error of the DBN-AR model (6) can be calculated as follows:

$$\xi(t) = y^{\circ}(t) - \widehat{y}(t), \quad t = n_y+1, n_y+2, \cdots, N \quad (9)$$

In this stage, the objective function for the parameter optimization of the DBN-AR model (6) is given as follows

$$\begin{aligned}E(t) &= \frac{1}{2}\xi^2(t) = \frac{1}{2}\left(y^{\circ}(t)-\widehat{y}(t)\right)^2\\ &= \frac{1}{2}\left(y^{\circ}(t)-\hat{\gamma}_0\left(\mathbf{\Theta}(t-1)\right)\right.\\ &\qquad \left.-\sum_{m=1}^{n_y}\hat{\gamma}_{y,m}\left(\mathbf{\Theta}(t-1)\right)y^{\circ}(t-m)\right)^2,\\ t &= n_y+1, n_y+2, \cdots, N\end{aligned} \quad (10)$$

where $y^{\circ}(t)$ is the actual value and $\widehat{y}(t)$ is the predicted value.

Using the normalized training data $\{y^{\circ}(t), t = 1, 2, \cdots, N\}$, all parameters of model (5) are fine-tuned by Eq. (A.20) (see Appendix). The predicted value from the DBN-AR model (5) in the fine-tuning stage is recalculated according to the following equation:

$$\tilde{y}_r(t) = \tilde{\gamma}_0\left(\mathbf{\Theta}(t-1)\right) + \sum_{m=1}^{n_y}\tilde{\gamma}_{y,m}\left(\mathbf{\Theta}(t-1)\right)y^{\circ}(t-m) \quad (11)$$

$$\begin{cases}\mathbf{M}_{n_y+i-1} = \left(1, y^{\circ}(n_y+i-1), y^{\circ}(n_y+i-2), \cdots, y^{\circ}(i)\right)\\ \Phi_{n_y+i-1} = \begin{pmatrix}\gamma_0\left(\mathbf{\Theta}(n_y+i-1)\right), & \gamma_{y,1}\left(\mathbf{\Theta}(n_y+i-1)\right),\\ \gamma_{y,2}\left(\mathbf{\Theta}(n_y+i-1)\right), \cdots, & \gamma_{y,n_y}\left(\mathbf{\Theta}(n_y+i-1)\right)\end{pmatrix}^{\mathrm{T}}\end{cases},$$

The parameter fine tuning process will not stop until the value of the mean square error, MSE $= \frac{1}{N-n_y}\sum_{t=n_y+1}^{N}\left(y^{\circ}(t)-\tilde{y}_r(t)\right)^2$ is smaller than a given value. Otherwise, the fine-tuning process will continue. After fine tuning stage, the final predicted value, $\tilde{y}_r(t)$, is denormalized to obtain the predicted output from the DBN-AR model (5), which is $\tilde{y}(t)$, using the denormalization formula, $\tilde{y}(t) = \min(y(t)) + \tilde{y}_r(t) * (\max(y(t)) - \min(y(t)))$.

IV. THE HYBRID PREDICTION MODEL

In this section, the proposed hybrid prediction model based on the EMD, DBN-AR model and BP neural network is used to forecast the time series data in this paper. The main structure of the hybrid prediction model is based on decomposition and ensemble, which is given in Fig. 2. The methods used in the proposed hybrid model are briefly introduced in the following.

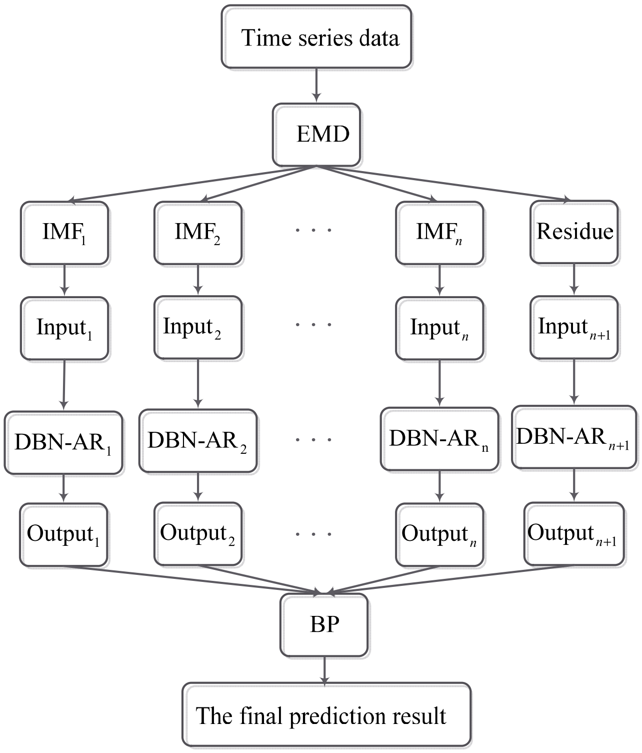

FIGURE 2. Schematic diagram of the proposed prediction model.

(1) Firstly, EMD is used to decompose the original time series into IMF components and one residual component.

(2) Secondly, use the DBN-AR model (5) to develop a prediction model for each extracted IMF component and the residual component, and then corresponding predictions are obtained for each component.

(3) Finally, the prediction results of all extracted IMFs and residual componets are obtained by DBN-AR model (5), and all the prediction results are combined to generate an output by a BP neural network, which can be used as the final prediction result for the original data series.

V. EVALUATION INDICATORS

To evaluate the efficiency of the hybrid model proposed in this paper, three evaluation indicators: the root mean square error (RMSE), the normalized mean squared error (NMSE) and the mean absolute percentage error (MAPE) are used, which are expressed as follows.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t=n_y}^{N}\left(c(t)-\breve{c}(t)\right)^2}{N}} \tag{12}$$

$$\mathrm{NMSE} = \frac{\sum_{t=n_y}^{N}\left(c(t)-\breve{c}(t)\right)^2}{\sum_{t=n_y}^{N}\left(c(t)-\bar{c}(t)\right)^2} \tag{13}$$

$$\mathrm{MAPE} = \frac{1}{N-n_y}\sum_{t=n_y+1}^{N}\left|\frac{c(t)-\breve{c}(t)}{c(t)} \times 100\%\right| \tag{14}$$

where $c(t)$, $\breve{c}(t)$ and $\bar{c}(t)$ is the actual value, predicted value and mean of the actual value, respectively. N is the length of the actual data series.

VI. EMPIRICAL STUDY

The validity of the proposed hybrid model is verified in this section, and a comprehensive experimental evaluation is proposed for the prediction model. Modelling problems for the PM2.5 levels in Beijing and and the weekly rates of British Pound/US dollar (GBP/USD) exchange rate data series are studied in this section, and the modelling results are obtained using a PC with an Intel(R) Core (TM) i7-9700 CPU @ 3.00GHz and MATLAB 2010b.

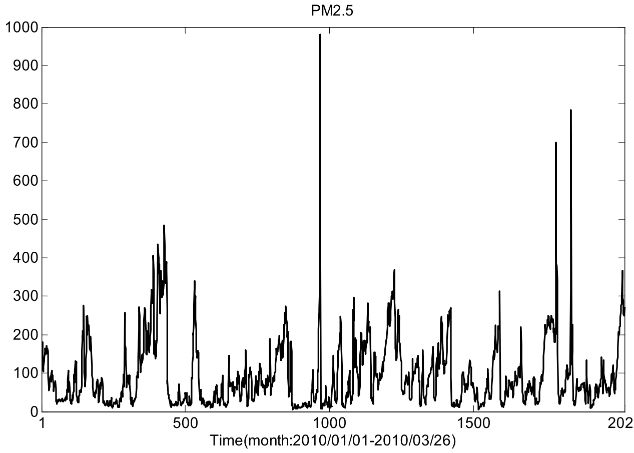

FIGURE 3. PM2.5 data series.

A. PM2.5 DATA SERIES

According to the PM2.5 meteorological records, the seriousness of PM2.5 in Beijing has attracted much attention [21], and many prediction methods have been proposed to predict the value of PM2.5 [22], [23], [24]. In this subsection, by learning the PM2.5 meteorological hour records from

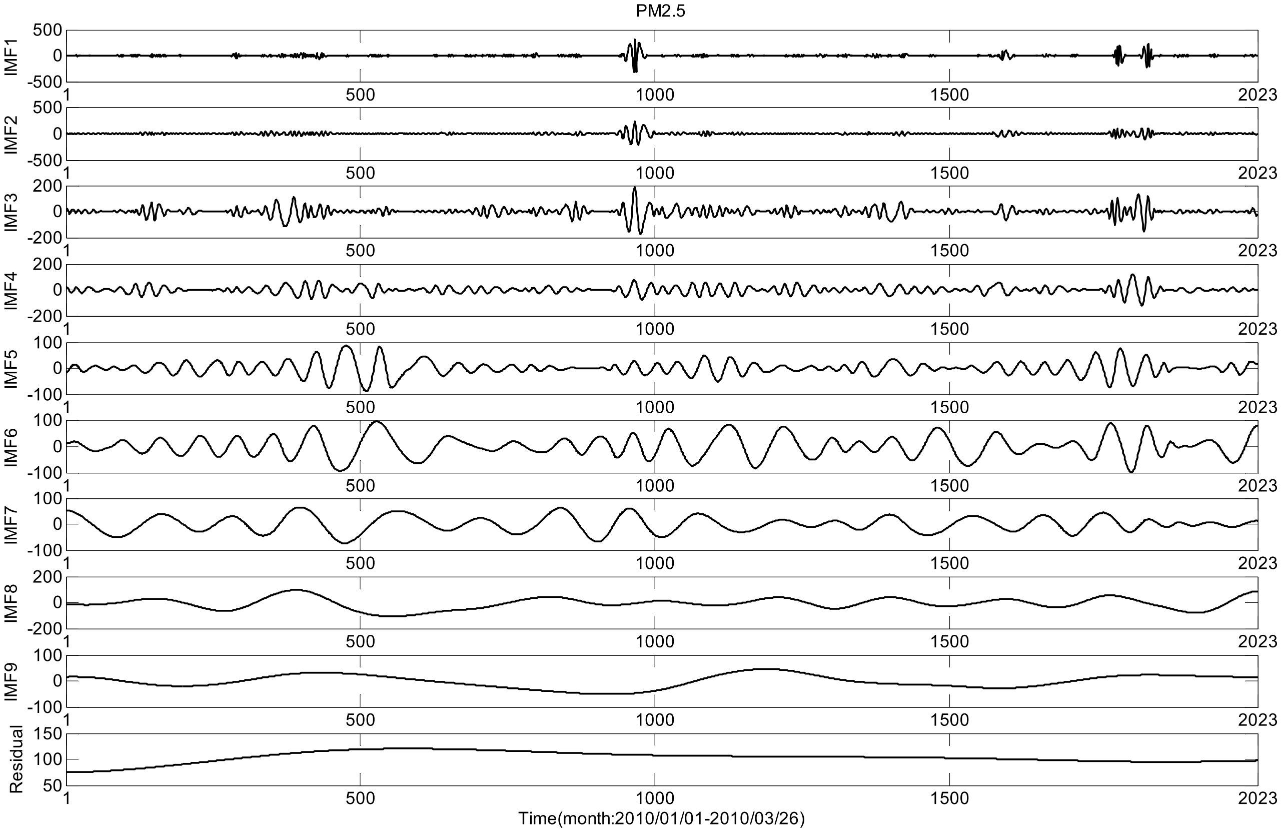

FIGURE 4. **The decomposed series of PM2.5 using EMD.**

1/1/2010 to 3/26/2010, we use the historical data series for PM2.5 concentrations (μg/m^3) to predict the future value of PM2.5 [25]. To facilitate PM2.5 data analysis, some abnormal values were removed. Therefore, a total of 2023 data series are obtained, as shown in Fig. 3.

The PM2.5 data series is a high-volatility and uncertain time series. Therefore, EMD is first used to decompose the original PM2.5 series, and the decomposition results are given in Fig. 4. In this paper, the original PM2.5 series is decomposed into 10 components in total, which are named IMF1, IMF2, ..., IMF9, residuals. Every subseries of the data series is composed of two sections: the first 1500 data points are used as training datasets, and the remaining 523 data points are used as testing datasets. To clearly demonstrate the prediction process for the proposed hybrid method, the single step ahead for forecasting is taken as an example in this section.

After the original PM2.5 data series are decomposed, the DBN-AR model is used to predict each IMF and residual. The input order for the DBN-AR model is chosen as $n_y = 5$, i.e., every five successive PM2.5 data series are applied to predict the sixth series using rolling technology. The number of iterations for each DBN module is 1000 in the pretraining stage, and the number of iterations of the DBN-AR model is chosen to be 2000 in the fine-tuning stage. The structure of each DBN module is chosen as $N_r^{(j)} = 2$; $Q_0^{(j)} = 5$; $Q_1^{(j)} = 15$; $Q_2^{(j)} = 1$; $(j = 0, 1, \cdots, 5)$. Based on these parameter settings, the DBN-AR model is used to predict each IMF component and the residue component. The constructed DBN-AR model is given in Eq. (15), as shown at the bottom of the next page. Figs. 5 and 6 show a comparison of the real values and predicted data for the training and testing dataset series, respectively. It can be observed from Figs. 5 and 6 that the predicted values for each IMF and the residue are almost consistent with the real values.

Next, the predicted results for all IMFs and a residue are combined by a BP neural network to obtain an output for the PM2.5 data series. The input value for the BP neural network is composed of all the IMFs and a residue, and the output value of the BP neural network is the PM2.5. Therefore, the numbers of input and output nodes are selected to be 10 and 1, respectively. After a series of experiments, the number of hidden nodes in this experiment was chosen to be 15. Finally, the predicted result for the proposed hybrid method is shown in Fig. 7 for the testing data series. It can be observed from Fig. 7 that the curves for the forecast value are very similar to the real value.

To verify the effectiveness of the proposed hybrid method for one-step ahead PM2.5 data series forecasting. The prediction results for the traditional AR model, DBN-AR model, EMD and DBN-AR method, and proposed hybrid method are given in Table 1 for comparison. In addition, two criteria,

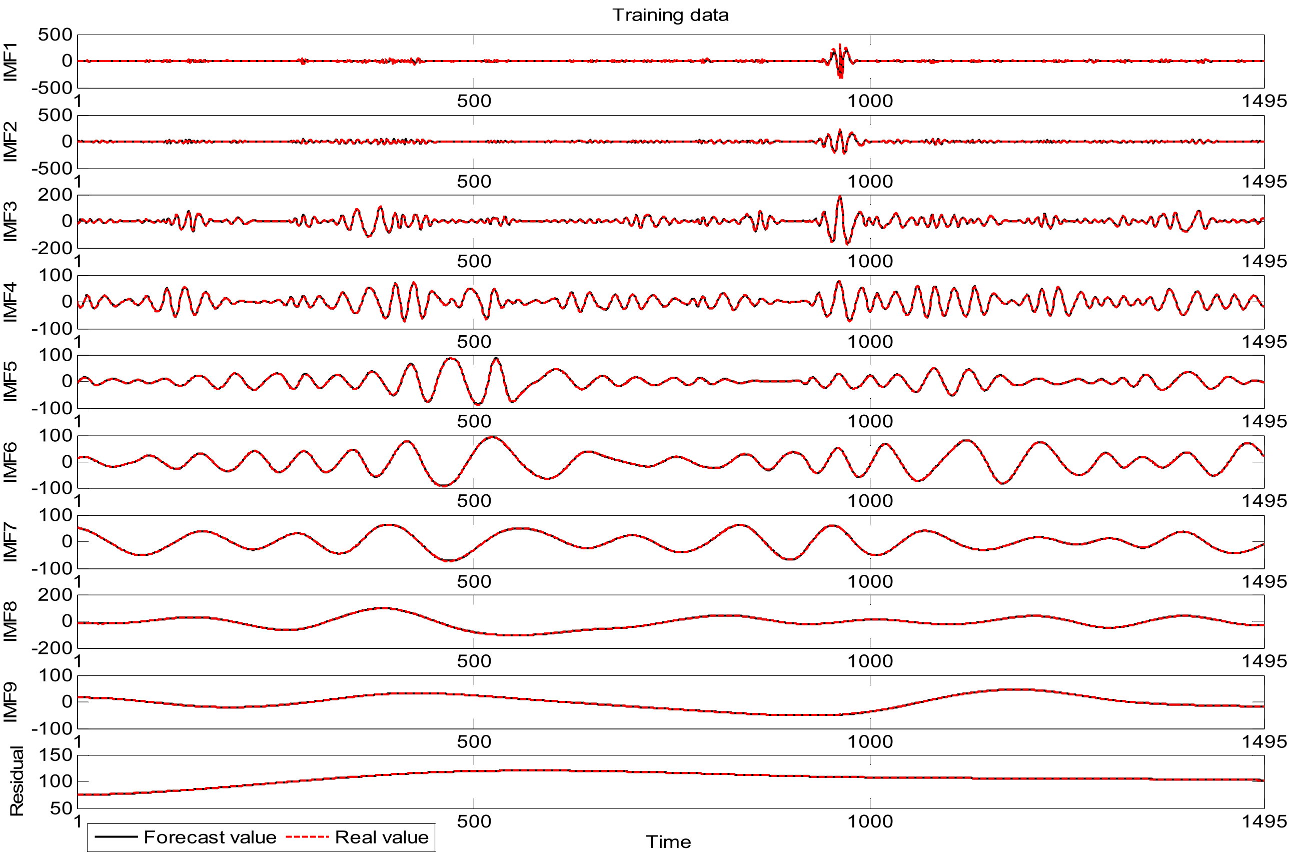

FIGURE 5. The predicted data and the real data for each IMF and a residue from the PM2.5 data series for the training data.

RMSE and NMSE, are employed to evaluate the performance of all prediction models. It can be observed from Table 1 that both the RMSE and NMSE values for the proposed hybrid method are small compared with all the other models, which confirms that the proposed method has better prediction accuracy than the other models.

B. FOREIGN CURRENCY EXCHANGE RATE DATA SERIES

The weekly rates of British Pound/US dollar (GBP/USD) exchange rate data are used to further verify the superiority of the proposed hybrid model. Data set contains 937 observations from the beginning of 1976 to the end of 1993 that is divided into 885 training data and 52 testing data sets.

$$\left\{\begin{aligned}
&y(t) = \gamma_0\left(\boldsymbol{\Theta}(t-1)\right) + \sum_{m=1}^{5} \gamma_{y,m}\left(\boldsymbol{\Theta}(t-1)\right) y(t-m) + \xi(t) \\
&\gamma_0\left(\boldsymbol{\Theta}(t-1)\right) = \varphi\left(u_{1,0}^{\left(N_r^{(0)}\right)}(t)\right) = \varphi\left(\mathbf{w}_{1,0}^{\left(N_r^{(0)}\right)} \mathbf{h}_0^{\left(N_r^{(0)}-1\right)}(t) + b_{1,0}^{\left(N_r^{(0)}\right)}\right) \\
&\gamma_{y,m}\left(\boldsymbol{\Theta}(t-1)\right) = \varphi\left(u_{1,m}^{\left(N_r^{(m)}\right)}(t)\right) = \varphi\left(\mathbf{w}_{1,m}^{\left(N_r^{(m)}\right)} \mathbf{h}_m^{\left(N_r^{(m)}-1\right)}(t) + b_{1,m}^{\left(N_r^{(m)}\right)}\right), \qquad m \in \{1, 2, \cdots, 5\} \\
&\mathbf{h}_j^{(\ell_j)}(t) = \left(h_{1,j}^{(\ell_j)}(t), h_{2,j}^{(\ell_j)}(t), \cdots, h_{Q_{\ell_j}^{(j)},j}^{(\ell_j)}(t)\right)^{\mathrm{T}}, \ell_j \in \left\{1, 2, \cdots, N_r^{(j)} - 1\right\}, \quad j \in \{0, 1, 2, \cdots, 5\} \\
&h_{n_{\ell_j}^{(j)},j}^{(\ell_j)}(t) = \varphi\left(u_{n_{\ell_j}^{(j)},j}^{(\ell_j)}(t)\right) = \varphi\left(\mathbf{w}_{n_{\ell_j}^{(j)},j}^{(\ell_j)} \mathbf{h}_j^{(\ell_j-1)}(t) + b_{n_{\ell_j}^{(j)},j}^{(\ell_j)}\right), \qquad n_{\ell_j}^{(j)} \in \left\{1, 2, \cdots, Q_{\ell_j}^{(j)}\right\} \\
&\mathbf{w}_{n_{\ell_j}^{(j)},j}^{(\ell_j)} = \left(w_{n_{\ell_j}^{(j)},1,j}^{(\ell_j)}, w_{n_{\ell_j}^{(j)},2,j}^{(\ell_j)}, \cdots, w_{n_{\ell_j}^{(j)},Q_{\ell_j-1}^{(j)},j}^{(\ell_j)}\right), \qquad Q_0^{(j)} = 5 \\
&\mathbf{h}_j^{(0)}(t) = \boldsymbol{\Theta}(t-1) = \left(y(t-1), y(t-2), \ldots, y(t-5)\right)^{\mathrm{T}}
\end{aligned}\right. \tag{15}$$

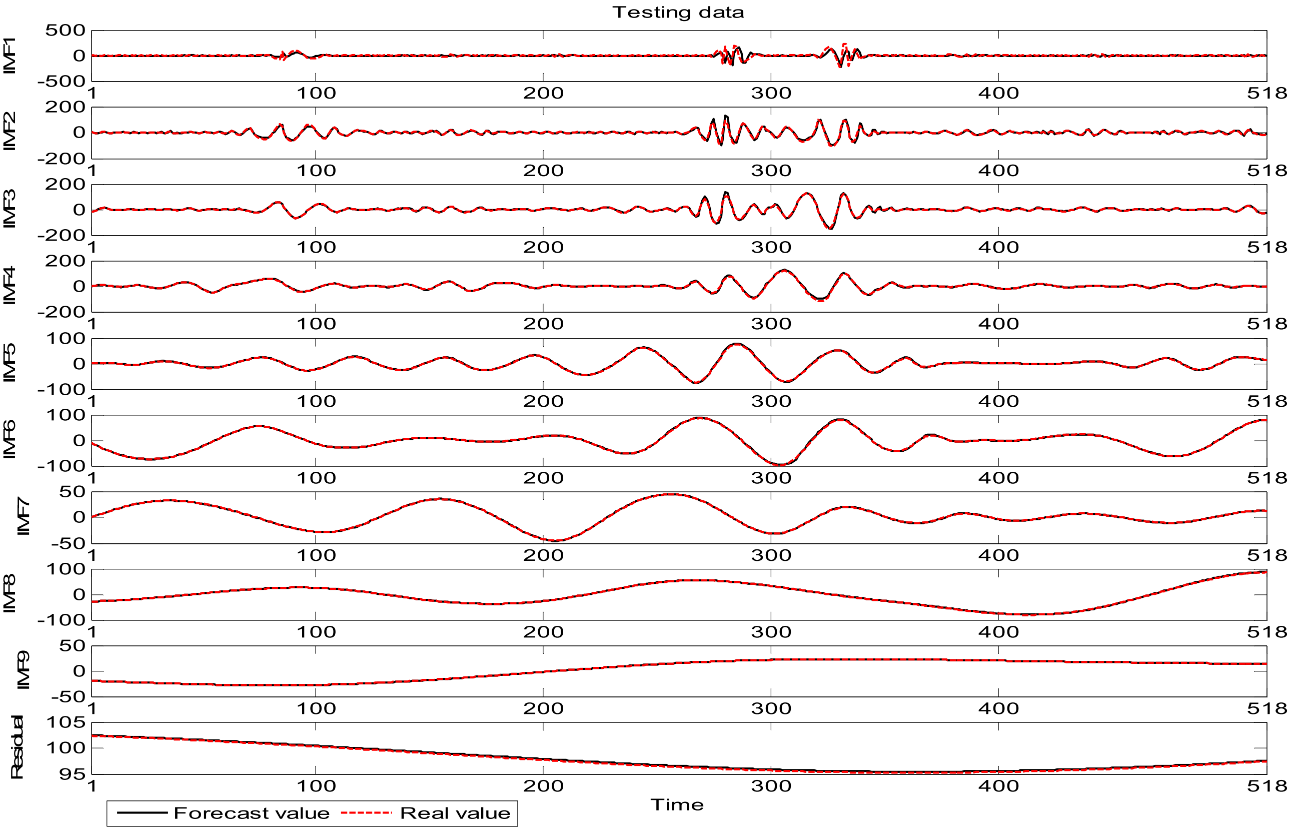

FIGURE 6. The predicted data and the real data for each IMF and a residue from the PM2.5 data series for the testing data.

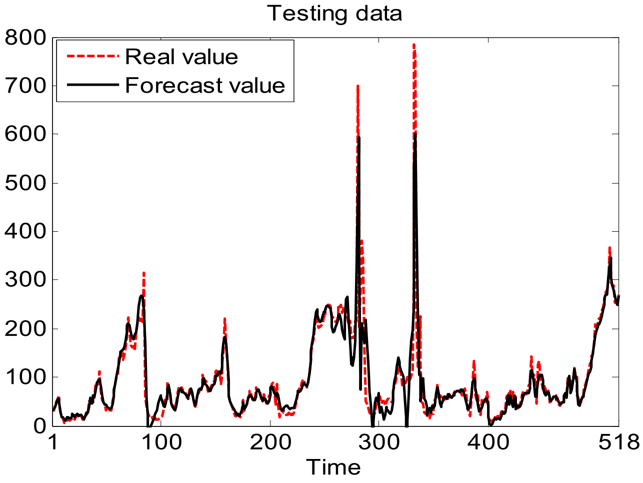

FIGURE 7. PM2.5 data forecasting results for the proposed method.

As done in [2] and [26], the input order of the proposed hybrid model is six. These data are obtained from the the database retrieval system of "Pacific Exchange Rate Service" (http://fx.sauder.ubc.ca/data.html) [26]. The modeling results of the proposed hybrid model and many other models from other literatures [2], [26] for the testing data are also given in Table 2. RMSE and MAPE values obtained from all testing data are given in Table 2. It can be seen from Table 2 that the prediction accuracy of the proposed hybrid model is better than that of the other models. Therefore, this a novel prediction model can provide an effective reference for the time series forecasting.

TABLE 1. Comparison of forecasting accuracy for testing data for different models.

Method	NMSE	RMSE
AR	0.532	44.565
DBN-AR	0.453	40.722
EMD-DBN-AR	0.384	34.544
Proposed	**0.378**	**33.976**

TABLE 2. Prediction results for weekly GB/USD exchange rate data.

Model	RMSE	MAPE
RW [26]	1.44×10^{-2}	1.60
ARMA [26]	1.38×10^{-2}	1.76
DBN(1) [26]	8.21×10^{-3}	0.98
DBN(2) [26]	7.69×10^{-3}	0.90
FFNN [26]	9.41×10^{-3}	1.11
DBN-AR [2]	7.30×10^{-3}	0.81
Proposed	$\mathbf{6.73\times10^{-3}}$	**0.79**

VII. CONCLUSION

In this paper, we propose a hybrid model for time series forecasting composed of EMD, DBN-AR and BP. The original time series signal is first decomposed into several IMFs and a residue by EMD, followed by a DBN-AR model which is used to model each extracted IMF and residue. Finally, the prediction results for all IMFs and a residue are combined by a BP to obtain an aggregated output for the time series data. Case studies indicate that the proposed hybrid model can be used to obtain better modelling accuracy compared to single and hybrid models.

APPENDIX
FINE TUNING PROCESS OF THE NONLINEAR DBN-AR MODEL [2]

Using formula (9), the objective function in the fine-tuning stage can be designed as follows

$$\begin{aligned} E(t) &= \frac{1}{2}\xi^2(t) = \frac{1}{2}\left(y^{\circ}(t) - \widehat{y}(t)\right)^2 \\ &= \frac{1}{2}\Bigg(y^{\circ}(t) - \hat{\gamma}_0\left(\mathbf{\Theta}(t-1)\right) \\ &\qquad - \sum_{m=1}^{n_y} \hat{\gamma}_{y,m}\left(\mathbf{\Theta}(t-1)\right) y^{\circ}(t-m)\Bigg)^2 \\ t &= n_y+1, n_y+2, \cdots, N \end{aligned} \tag{A.1}$$

where $y^{\circ}(t)$ is the actual value and $\widehat{y}(t)$ is the prediction value of the DBN-AR model.

All the parameters in the DBN-AR model are fine tuned by the specially designed gradient descent method. For the neuron in the last output layer, which is the $N_r^{(j)}$-th layer, Eq. (10) is used to update the gradient for the parameter.

$$\begin{aligned} \frac{\partial E(t)}{\partial w^{\left(N_r^{(j)}\right)}_{1,n^{(j)}_{N_r^{(j)}-1},j}} &= \frac{\partial E(t)}{\partial \xi(t)} \frac{\partial \xi(t)}{\partial \widehat{y}(t)} \frac{\partial \widehat{y}(t)}{\partial \hat{\gamma}_{y,m}} \frac{\partial \hat{\gamma}_{y,m}}{\partial u^{\left(N_r^{(j)}\right)}_{1,j}(t)} \frac{\partial u^{\left(N_r^{(j)}\right)}_{1,j}(t)}{\partial w^{\left(N_r^{(j)}\right)}_{1,n^{(j)}_{N_r^{(j)}-1},j}} \\ &= -\xi(t)\, a(t-j)\, \varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}(t)\right) h^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t) \\ &= \varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}(t)\right) c(t-j)\, h^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t) \end{aligned} \tag{A.2}$$

where

$$\begin{aligned} a(t-j) &= y^{\circ}(t-j), j = 1, 2, \cdots, n_y;\ a(t) = 1 \\ c(t-j) &= -\xi(t)\, a(t-j),\ j = 0, 1, 2, \cdots, n_y \end{aligned}$$

and $\varphi'(u)$ represents the derivative of $\varphi(u)$ with respect to u. Let

$$\delta^{\left(N_r^{(j)}\right)}_{1,j}(t) = \varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\right) c(t-j) \tag{A.3}$$

which is the local gradient with respect to neuron of the last layer in the j-th DBN module. Therefore, Eq. (A.2) can be rewritten as

$$\frac{\partial E(t)}{\partial w^{\left(N_r^{(j)}\right)}_{1,n^{(j)}_{N_r^{(j)}-1},j}} = \delta^{\left(N_r^{(j)}\right)}_{1,j}(t)\, h^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t) \tag{A.4}$$

Similarly, one can obtain

$$\begin{aligned} \frac{\partial E(t)}{\partial b^{\left(N_r^{(j)}\right)}_{1,j}} &= \frac{\partial E(t)}{\partial \xi(t)} \frac{\partial \xi(t)}{\partial \widehat{y}(t)} \frac{\partial \widehat{y}(t)}{\partial \hat{\gamma}_{y,m}(t)} \frac{\partial \hat{\gamma}_{y,m}(t)}{\partial u^{\left(N_r^{(j)}\right)}_{1,j}(t)} \frac{\partial u^{\left(N_r^{(j)}\right)}_{1,j}(t)}{\partial b^{\left(N_r^{(j)}\right)}_{1,j}} \\ &= -\xi(t)\, a(t-j)\, \varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}(t)\right) \\ &= \varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\right) c(t-j) \\ &= \delta^{\left(N_r^{(j)}\right)}_{1,j}(t) \end{aligned} \tag{A.5}$$

For neuron $n^{(j)}_{N_r^{(j)}-1} \in \left\{1, 2, \cdots, Q^{(j)}_{N_r^{(j)}-1}\right\}$ in the $(N_r^{(j)}-1)$-th layer, Eq. (A.6) is used to update the gradient for the parameter, (A.6), as shown at the bottom of the page.

Let

$$\delta^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t) = \varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t)\right) w^{\left(N_r^{(j)}\right)}_{1,n^{(j)}_{N_r^{(j)}-1},j}\, \delta^{\left(N_r^{(j)}\right)}_{1,j}(t) \tag{A.7}$$

$$\begin{aligned} \frac{\partial E(t)}{\partial w^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},n^{(j)}_{N_r^{(j)}-2},j}} &= \frac{\partial E(t)}{\partial \xi(t)} \frac{\partial \xi(t)}{\partial \widehat{y}(t)} \frac{\partial \widehat{y}(t)}{\partial \hat{\gamma}_{y,m}(t)} \frac{\partial \hat{\gamma}_{y,m}(t)}{\partial u^{\left(N_r^{(j)}\right)}_{1,j}(t)} \frac{\partial u^{\left(N_r^{(j)}\right)}_{1,j}(t)}{\partial h^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t)} \frac{\partial h^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t)}{\partial u^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t)} \frac{\partial u^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t)}{\partial w^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},n^{(j)}_{N_r^{(j)}-2},j}} \\ &= -\xi(t)\, a(t-j)\, \varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}(t)\right) w^{\left(N_r^{(j)}\right)}_{1,n^{(j)}_{N_r^{(j)}-1},j}\, \varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t)\right) h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t) \\ &= \varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t)\right) w^{\left(N_r^{(j)}\right)}_{1,n^{(j)}_{N_r^{(j)}-1},j}\, \varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}(t)\right) c(t-j)\, h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t) \\ &= \varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}(t)\right) w^{\left(N_r^{(j)}\right)}_{1,n^{(j)}_{N_r^{(j)}-1},j}\, \delta^{\left(N_r^{(j)}\right)}_{1,j}(t)\, h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t) \end{aligned} \tag{A.6}$$

$$
\begin{aligned}
\frac{\partial E\left(t\right)}{\partial b^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}}
&= \frac{\partial E\left(t\right)}{\partial \xi\left(t\right)}\frac{\partial \xi\left(t\right)}{\partial \widehat{y}\left(t\right)}\frac{\partial \widehat{y}\left(t\right)}{\partial \hat{\gamma}_{y,m}\left(t\right)}\frac{\partial \hat{\gamma}_{y,m}\left(t\right)}{\partial u^{\left(N_r^{(j)}\right)}_{1,j}\left(t\right)}\frac{\partial u^{\left(N_r^{(j)}\right)}_{1,j}\left(t\right)}{\partial h^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}\left(t\right)}\frac{\partial h^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}\left(t\right)}{\partial u^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}\left(t\right)}\frac{\partial u^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}\left(t\right)}{\partial b^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}} \\
&= -\xi\left(t\right)a\left(t-j\right)\varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\left(t\right)\right)w^{\left(N_r^{(j)}\right)}_{1,n^{(j)}_{N_r^{(j)}-1},j}\varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}\left(t\right)\right) \\
&= \varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}\left(t\right)\right)w^{\left(N_r^{(j)}\right)}_{1,n^{(j)}_{N_r^{(j)}-1},j}\varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\left(t\right)\right)c\left(t-j\right) \\
&= \delta^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}\left(t\right) \qquad \text{(A.9)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial E(t)}{\partial w^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},n^{(j)}_{N_r^{(j)}-3},j}}
&= \frac{\partial E(t)}{\partial \xi(t)}\frac{\partial \xi(t)}{\partial \widehat{y}(t)}\frac{\partial \widehat{y}(t)}{\partial \hat{\gamma}_{y,m}(t)}
\left(\begin{array}{l}
\frac{\partial \hat{\gamma}_{y,m}(t)}{\partial h^{\left(N_r^{(j)}-1\right)}_{1,j}(t)}\frac{\partial h^{\left(N_r^{(j)}-1\right)}_{1,j}(t)}{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}\frac{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}\frac{\partial u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial w^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},n^{(j)}_{N_r^{(j)}-3},j}} \\
+\frac{\partial \hat{\gamma}_{y,m}(t)}{\partial h^{\left(N_r^{(j)}-1\right)}_{2,j}(t)}\frac{\partial h^{\left(N_r^{(j)}-1\right)}_{2,j}(t)}{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}\frac{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial u^{N_r^{j}-2}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}\frac{\partial u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial w^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},n^{(j)}_{N_r^{(j)}-3},j}} \\
+\cdots \\
+\frac{\partial \hat{\gamma}_{y,m}(t)}{\partial h^{\left(N_r^{(j)}-1\right)}_{Q^{(j)}_{N_r^{(j)}-1},j}(t)}\frac{\partial h^{\left(N_r^{(j)}-1\right)}_{Q^{(j)}_{N_r^{(j)}-1},j}(t)}{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}\frac{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}\frac{\partial u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial w^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},n^{(j)}_{N_r^{(j)}-3},j}}
\end{array}\right) \\
&= -\xi(t)a\left(t-j\right)
\left(\begin{array}{l}
\varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\right)w^{\left(N_r^{(j)}\right)}_{1,1,j}\varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{1,j}\right)w^{\left(N_r^{(j)}-1\right)}_{1,n^{(j)}_{N_r^{(j)}-2},j}\varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right)h^{\left(N_r^{(j)}-3\right)}_{n^{(j)}_{N_r^{(j)}-3},j}\left(t\right) \\
+\varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\right)w^{\left(N_r^{(j)}\right)}_{1,2,j}\varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{2,j}\right)w^{\left(N_r^{(j)}-1\right)}_{2,n^{(j)}_{N_r^{(j)}-2},j}\varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right)h^{\left(N_r^{(j)}-3\right)}_{n^{(j)}_{N_r^{(j)}-3},j}\left(t\right) \\
+\cdots \\
+\varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\right)w^{\left(N_r^{(j)}\right)}_{1,Q^{(j)}_{N_r^{(j)}-1},j}\varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{Q^{(j)}_{N_r^{(j)}-1},j}\right)w^{\left(N_r^{(j)}-1\right)}_{Q^{(j)}_{N_r^{(j)}-1},n^{(j)}_{N_r^{(j)}-2},j}\varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right)h^{\left(N_r^{(j)}-3\right)}_{n^{(j)}_{N_r^{(j)}-3},j}\left(t\right)
\end{array}\right) \\
&= \sum_{v=1}^{Q^{(j)}_{N_r^{(j)}-1}}\varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right)w^{\left(N_r^{(j)}-1\right)}_{v,n^{(j)}_{N_r^{(j)}-2},j}\varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{v,j}\right)w^{\left(N_r^{(j)}\right)}_{1,v,j}\varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\right)c\left(t-j\right)h^{\left(N_r^{(j)}-3\right)}_{n^{(j)}_{N_r^{(j)}-3},j}\left(t\right) \\
&= \sum_{v=1}^{Q^{(j)}_{N_r^{(j)}-1}}\varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right)w^{\left(N_r^{(j)}-1\right)}_{v,n^{(j)}_{N_r^{(j)}-2},j}\delta^{\left(N_r^{(j)}-1\right)}_{v,j}\left(t\right)h^{\left(N_r^{(j)}-3\right)}_{n^{(j)}_{N_r^{(j)}-3},j}\left(t\right) \qquad \text{(A.10)}
\end{aligned}
$$

Then, Eq. (A.6) becomes

$$\frac{\partial E\left(t\right)}{\partial w^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},n^{(j)}_{N_r^{(j)}-2},j}} = \delta^{\left(N_r^{(j)}-1\right)}_{n^{(j)}_{N_r^{(j)}-1},j}\left(t\right) h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\left(t\right) \tag{A.8}$$

Similarly, one can obtain, (A.9), as shown at the previous page.

For neuron $n^{(j)}_{N_r^{(j)}-2} \in \left\{1, 2, \cdots, Q^{(j)}_{N_r^{(j)}-2}\right\}$ in the $(N_r^{(j)}-2)$-th layer, Eq. (A.10) can be used to update the gradient for the parameter, (A.10), as shown at the previous page.

Let

$$\delta^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\left(t\right) = \sum_{v=1}^{Q^{(j)}_{N_r^{(j)}-1}} \varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right) w^{\left(N_r^{(j)}-1\right)}_{v,n^{(j)}_{N_r^{(j)}-2},j} \delta^{\left(N_r^{(j)}-1\right)}_{v,j}\left(t\right) \tag{A.11}$$

Then, Eq. (A.10) becomes

$$\frac{\partial E\left(t\right)}{\partial w^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},n^{(j)}_{N_r^{(j)}-3},j}} = \delta^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\left(t\right) h^{\left(N_r^{(j)}-3\right)}_{n^{(j)}_{N_r^{(j)}-3},j}\left(t\right) \tag{A.12}$$

Similarly, the following equation can be obtained as in (A.13), shown at the bottom of the page.

Therefore, the local gradient for each neuron of layer ℓ_j in the j-th DBN module can be obtained by the derivation process detailed above and computed using the following equation:

$$\delta^{(\ell_j)}_{n^{(j)}_{\ell_j},j}\left(t\right) = \sum_{v=1}^{Q^{(j)}_{\ell_j+1}} \varphi'\left(u^{(\ell_j)}_{n^{(j)}_{\ell_j},j}\right) w^{(\ell_j+1)}_{v,n^{(j)}_{\ell_j},j} \delta^{(\ell_j+1)}_{v,j}\left(t\right), \quad \ell_j \in \left\{1, 2, \cdots, N_r^{(j)}-2\right\} \tag{A.14}$$

$$\begin{aligned}
\frac{\partial E(t)}{\partial b^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}} &= \frac{\partial E(t)}{\partial \xi(t)} \frac{\partial \xi(t)}{\partial \widehat{y}(t)} \frac{\partial \widehat{y}(t)}{\partial \hat{\gamma}_{y,m}(t)} \left(\begin{array}{l}
\dfrac{\partial \hat{\gamma}_{y,m}(t)}{\partial h^{\left(N_r^{(j)}-1\right)}_{1,j}(t)} \dfrac{\partial h^{\left(N_r^{(j)}-1\right)}_{1,j}(t)}{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)} \dfrac{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)} \dfrac{\partial u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial b^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}} \\
+\dfrac{\partial \hat{\gamma}_{y,m}(t)}{\partial h^{\left(N_r^{(j)}-1\right)}_{2,j}(t)} \dfrac{\partial h^{\left(N_r^{(j)}-1\right)}_{2,j}(t)}{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)} \dfrac{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial u^{N_r^{j}-2}_{n^{(j)}_{N_r^{(j)}-2},j}(t)} \dfrac{\partial u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial b^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}} \\
+\cdots \\
+\dfrac{\partial \hat{\gamma}_{y,m}(t)}{\partial h^{\left(N_r^{(j)}-1\right)}_{Q^{(j)}_{N_r^{(j)}-1},j}(t)} \dfrac{\partial h^{\left(N_r^{(j)}-1\right)}_{Q^{(j)}_{N_r^{(j)}-1},j}(t)}{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)} \dfrac{\partial h^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)} \dfrac{\partial u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}(t)}{\partial b^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}}
\end{array}\right) \\
&= -\xi(t)a\left(t-j\right) \left(\begin{array}{l}
\varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\right) w^{\left(N_r^{(j)}\right)}_{1,1,j} \varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{1,j}\right) w^{\left(N_r^{(j)}-1\right)}_{1,n^{(j)}_{N_r^{(j)}-2},j} \varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right) \\
+\varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\right) w^{\left(N_r^{(j)}\right)}_{1,2,j} \varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{2,j}\right) w^{\left(N_r^{(j)}-1\right)}_{2,n^{(j)}_{N_r^{(j)}-2},j} \varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right) \\
+\cdots \\
+\varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\right) w^{\left(N_r^{(j)}\right)}_{1,Q^{(j)}_{N_r^{(j)}-1},j} \varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{Q^{(j)}_{N_r^{(j)}-1},j}\right) w^{\left(N_r^{(j)}-1\right)}_{Q^{(j)}_{N_r^{(j)}-1},n^{(j)}_{N_r^{(j)}-2},j} \varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right)
\end{array}\right) \\
&= \sum_{v=1}^{Q^{(j)}_{N_r^{(j)}-1}} \varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right) w^{\left(N_r^{(j)}-1\right)}_{v,n^{(j)}_{N_r^{(j)}-2},j} \varphi'\left(u^{\left(N_r^{(j)}-1\right)}_{v,j}\right) w^{\left(N_r^{(j)}\right)}_{1,v,j} \varphi'\left(u^{\left(N_r^{(j)}\right)}_{1,j}\right) c\left(t-j\right) \\
&= \sum_{v=1}^{Q^{(j)}_{N_r^{(j)}-1}} \varphi'\left(u^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\right) w^{\left(N_r^{(j)}-1\right)}_{v,n^{(j)}_{N_r^{(j)}-2},j} \delta^{\left(N_r^{(j)}-1\right)}_{v,j}\left(t\right) \\
&= \delta^{\left(N_r^{(j)}-2\right)}_{n^{(j)}_{N_r^{(j)}-2},j}\left(t\right)
\end{aligned} \tag{A.13}$$

$$\begin{cases} w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j}(k) \Leftarrow w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j}(k-1) + \Delta w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j}(k) + \alpha\left(w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j}(k-1) - w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j}(k-2)\right) \\ b^{(L)}_{n_L^{(j)},j}(k) \Leftarrow b^{(L)}_{n_L^{(j)},j}(k-1) + \Delta b^{(L)}_{n_L^{(j)},j}(k) + \alpha\left(b^{(L)}_{n_L^{(j)},j}(k-1) - b^{(L)}_{n_L^{(j)},j}(k-2)\right) \end{cases} \tag{A.20}$$

and the gradients to the connection weight and bias can be calculated using Eqs. (A. 15) and (A. 16), respectively.

$$\frac{\partial E(t)}{\partial w^{(\ell_j)}_{n_{\ell_j}^{(j)},n_{\ell_j-1}^{(j)},j}} = \delta^{(\ell_j)}_{n_{\ell_j}^{(j)},j}(t)\, h^{(\ell_j-1)}_{n_{\ell_j-1}^{(j)},j}(t) \tag{A.15}$$

$$\frac{\partial E(t)}{\partial b^{(\ell_j)}_{n_{\ell_j}^{(j)},j}} = \delta^{(\ell_j)}_{n_{\ell_j}^{(j)},j}(t) \tag{A.16}$$

When all the gradients are calculated by using Eqs. (A.3-A.5), (A.7-A.9), and (A.11-A.16), the following parameter updating values are obtained

$$\Delta w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j} = -\eta \frac{\partial E(t)}{\partial w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j}} = -\eta \delta^{(L)}_{n_L^{(j)},j} h^{(L-1)}_{n_{L-1}^{(j)},j}(t) \tag{A.17}$$

$$\Delta b^{(L)}_{n_L^{(j)},j} = -\eta \frac{\partial E(t)}{\partial b^{(L)}_{n_L^{(j)},j}} = -\eta \delta^{(L)}_{n_L^{(j)},j} \tag{A.18}$$

where $L \in \left\{1, 2, \cdots, N_r^{(j)} - 1, N_r^{(j)}\right\}$ and $\eta > 0$ is the predetermined learning rate. Then, the parameters are updated using

$$\begin{cases} w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j} \Leftarrow w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j} + \Delta w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j} \\ b^{(L)}_{n_L^{(j)},j} \Leftarrow b^{(L)}_{n_L^{(j)},j} + \Delta b^{(L)}_{n_L^{(j)},j} \end{cases} \tag{A.19}$$

where the initial values of $w^{(L)}_{n_L^{(j)},n_{L-1}^{(j)},j}$ and $b^{(L)}_{n_L^{(j)},j}$ are calculated in the pretraining stage of the DBN-AR model estimation. Furthermore, to avoid parameter oscillation in the fine-tuning process and to slow down the convergence rate, the momentum term is added to the final parameter updating rule as in (A.20), shown at the top of the page, where k denotes the number of parameter updating, iterations and $\alpha \in [0, 1)$ represents the predetermined momentum factor.

REFERENCES

[1] C. N. Babu and B. E. Reddy, "A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data," *Appl. Soft Comput.*, vol. 23, pp. 27–38, Oct. 2014.

[2] W. Xu, H. Peng, X. Zeng, F. Zhou, X. Tian, and X. Peng, "Deep belief network-based AR model for nonlinear time series forecasting," *Appl. Soft Comput.*, vol. 77, pp. 605–621, Apr. 2019.

[3] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," *Renew. Sustain. Energy Rev.*, vol. 13, no. 4, pp. 915–920, 2009.

[4] Z. Du, "Forecasting the daily 10.7 cm solar radio flux using an autoregressive model," *Sol. Phys.*, vol. 295, no. 9, pp. 1–23, Sep. 2020.

[5] H. Zhang, S. Zhang, P. Wang, Y. Qin, and H. Wang, "Forecasting of particulate matter time series using wavelet analysis and wavelet-ARMA/ARIMA model in Taiyuan, China," *J. Air Waste Manage. Assoc.*, vol. 67, no. 7, pp. 776–788, 2017.

[6] A. S. Khaniani, H. Motieyan, and A. Mohammadi, "Rainfall forecast based on GPS PWV together with meteorological parameters using neural network models," *J. Atmos. Solar-Terr. Phys.*, vol. 214, Mar. 2021, Art. no. 105533.

[7] L. F. S. Vilela, R. C. Leme, C. A. M. Pinheiro, and O. A. S. Carpinteiro, "Forecasting financial series using clustering methods and support vector regression," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 743–773, Aug. 2019.

[8] C. H. Hu, H. Pei, X.-S. Si, D.-B. Du, Z.-N. Pang, and X. Wang, "A prognostic model based on DBN and diffusion process for degrading bearing," *IEEE Trans. Ind. Electron.*, vol. 67, no. 10, pp. 8767–8777, Oct. 2019.

[9] G. Shan, H. Wang, W. Liang, and K. Chen, "Robust encoder–decoder learning framework for offline handwritten mathematical expression recognition based on a multi-scale deep neural network," *Sci. China Inf. Sci.*, vol. 64, no. 3, Mar. 2021, Art. no. 139101.

[10] X. Xiao, H. Mo, Y. Zhang, and G. Shan, "Meta-ANN—A dynamic artificial neural network refined by meta-learning for short-term load forecasting," *Energy*, vol. 246, May 2022, Art. no. 123418.

[11] M. Abdollahzade, A. Miranian, H. Hassani, and H. Iranmanesh, "A new hybrid enhanced local linear neuro-fuzzy model based on the optimized singular spectrum analysis and its application for nonlinear and chaotic time series forecasting," *Inf. Sci.*, vol. 295, pp. 107–125, Feb. 2015.

[12] Z. Diao, D. Zhang, X. Wang, K. Xie, S. He, X. Lu, and Y. Li, "A hybrid model for short-term traffic volume prediction in massive transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 935–946, Mar. 2019.

[13] P. Singh and P. Dwivedi, "A novel hybrid model based on neural network and multi-objective optimization for effective load forecast," *Energy*, vol. 182, pp. 606–622, Sep. 2019.

[14] M. Niu, Y. Hu, S. Sun, and Y. Liu, "A novel hybrid decomposition-ensemble model based on VMD and HGWO for container throughput forecasting," *Appl. Math. Model.*, vol. 57, pp. 163–178, May 2018.

[15] J. Li, Q. Wu, Y. Tian, and L. Fan, "Monthly Henry hub natural gas spot prices forecasting using variational mode decomposition and deep belief network," *Energy*, vol. 227, Jul. 2021, Art. no. 120478.

[16] D. Wang, H. Guo, H. Luo, O. Grunder, Y. Lin, and H. Guo, "Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm," *Appl. Energy*, vol. 190, pp. 390–407, Mar. 2017.

[17] X. Qiu, P. N. Suganthan, and G. A. J. Amaratunga, "Ensemble incremental learning random vector functional link network for short-term electric load forecasting," *Knowl.-Based Syst.*, vol. 145, pp. 182–196, Apr. 2018.

[18] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London A, Math., Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

[19] G. E. Hinton and T. J. Sejnowski, *Learning and Relearning in Boltzmann Machines*. Cambridge, MA, USA: MIT Press, 2001.

[20] G. Fu, "Deep belief network based ensemble approach for cooling load forecasting of air-conditioning system," *Energy*, vol. 148, pp. 269–282, Apr. 2018.

[21] X. Liang, T. Zou, B. Guo, S. Li, H. Zhang, S. Zhang, H. Huang, andS. X. Chen, "Assessing Beijing's PM$_{2.5}$ pollution: Severity, weather impact, APEC and winter heating," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 471, p. 2182, Oct. 2015.

[22] X. Lu, Y. H. Sha, Z. Li, Y. Huang, W. Chen, D. Chen, J. Shen, Y. Chen, and J. C. H. Fung, "Development and application of a hybrid long-short term memory—Three dimensional variational technique for the improvement of PM$_{2.5}$ forecasting," *Sci. Total Environ.*, vol. 770, May 2021, Art. no. 144221.

[23] A. Sayeed, Y. Lops, Y. Choi, J. Jung, and A. K. Salman, "Bias correcting and extending the PM forecast by CMAQ up to 7 days using deep convolutional neural networks," *Atmos. Environ.*, vol. 253, May 2021, Art. no. 118376.

[24] E. Sharma, R. C. Deo, R. Prasad, and A. V. Parisi, "A hybrid air quality early-warning framework: An hourly forecasting model with online sequential extreme learning machines and empirical mode decomposition algorithms," *Sci. Total Environ.*, vol. 709, Mar. 2020, Art. no. 135934.

[25] X. Liang, T. Zou, B. Guo, S. Li, H. Zhang, S. Zhang, H. Huang, and S. X. Chen, "Assessing Beijing's PM$_{2.5}$ pollution: Severity, weather impact, APEC and winter heating," *Proc. Roy. Soc. A*, vol. 471, Oct. 2015, Art. no. 20150257. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data

[26] F. Shen, J. Chao, and J. Zhao, "Forecasting exchange rate using deep belief networks and conjugate gradient method," *Neurocomputing*, vol. 167, pp. 243–253, Nov. 2015.

HUI HU received the M.S. degree in applied psychology from Anhui Normal University, Wuhu, China, in 2011.

She is currently a Lecturer with the School of Teacher Education, Anqing Normal University, Anqing, China. Her current research interests include psychometrics and education statistics, artificial neural networks, and time series forecast.

WENQUAN XU received the M.S. degree in control theory and control engineering from Hohai University, Nanjing, China, in 2011, and the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2021.

He is currently an Associate Professor with the School of Electronic Engineering and Intelligent Manufacturing, Anqing Normal University, Anqing, China. His current research interests include deep learning, system identification, and nonlinear time series analysis.